IN THE CLAIMS

Please amend the claims as follows:

Claims 1-9 (Canceled).

Claim 10 (Currently Amended): A method of producing a system architecture with a design tool computer, the system architecture including a plurality of electrical components connected to each other, the components including electronic control units, sensors and actuators, the method comprising:

a) identifying a set of undesirable events and ascribing to each of the undesirable events an indicator of ~~their~~ severity;

b) associating ~~where possible~~ each of the undesirable events with ~~one or more~~ any involved actuator of the actuators of the system architecture;

c) developing a functional specification of an initial architecture proposed for implementation of the system architecture, the functional specification of the initial architecture including dataflow for and between electrical components thereof;

d) refining on the functional specification fault tolerance requirements associated with the severity of each of the undesirable events and issuing refined fault tolerance requirements of the functional specification;

e) producing replicates in the functional specification together with attached indicators of ~~independence~~ freeness of the replicates from other of the replicates, the indicators reflecting the refined fault tolerance requirements;

f) defining a hardware structure for the system architecture;

g) mapping, via the design tool computer, the functional specification onto the hardware structure; and

h) verifying automatically that the indicators of ~~independence~~ freeness are preserved during the mapping.

Claim 11 (Previously Presented): A method according to claim 10, wherein the system includes a fault tolerant system.

Claim 12 (Currently Amended): A method according to claim 10, ~~including, in~~ wherein the developing (c)[[,]] includes defining a series of modes of operation.

Claim 13 (Previously Presented): A method according to claim 12, wherein the modes of operation include nominal and limp-home modes.

Claim 14 (Currently Amended): A method according to claim 12, ~~including~~ wherein the defining a series of modes of operation includes specifying the series of modes in a form of one or more state charts.

Claim 15 (Currently Amended): A method according to claim 10, further including mapping geometrically hardware components and/or wiring and then verifying automatically that the indicators of ~~independence~~ freeness are preserved by the geometrical mapping.

Claim 16 (Previously Presented): A method according to claim 10, further including specifying the severity in a form of probability of failure per unit of time.

Claim 17 (Previously Presented): A method according to claim 10, further including outputting a set of data for use in manufacturing the system architecture.

Claim 18 (Previously Presented): A method according to claim 10, wherein the system architecture includes a safety critical architecture for a vehicle.

Claim 19 (Previously Presented): A method according to claim 10, wherein the hardware structure is in a form of a series of electronic control units connected to each other by networks.

Claim 20 (Currently Amended): A ~~computer program product comprising a~~ computer readable storage medium having thereon computer ~~program code means~~ executable instructions that, when ~~the program is loaded~~ executed by a computer, [[to]] make the computer execute a ~~procedure~~ method to design and verify a system architecture, the ~~procedure~~ method comprising:

a) identifying a set of undesirable events and ascribing to each of the undesirable events an indicator of ~~their~~ severity;

b) associating ~~where possible~~ each of the undesirable events with ~~one or more~~ any involved actuator of the actuators of the system architecture;

c) developing a functional specification of an initial architecture proposed for implementation of the system architecture, the functional specification of the initial architecture including dataflow for and between components thereof;

d) refining on the functional specification fault tolerance requirements associated with the severity of each of the undesirable events and issuing refined fault tolerance requirements of the functional specification;

e) producing replicates in the functional specification together with attached indicators of ~~independence~~ freeness of the replicates from other of the replicates, the indicators reflecting the refined fault tolerance requirements;

f) defining a hardware structure for the system architecture;

g) mapping the functional specification onto the hardware structure; and

h) verifying automatically that the indicators of ~~independence~~ freeness are preserved during the mapping.

Claim 21 (Currently Amended): A ~~method~~ computer readable storage medium according to claim 20, wherein the hardware structure is in a form of a series of electronic control units connected to each other by networks.

Claim 22 (Currently Amended): A computer ~~program product~~ readable storage medium according to claim 20, wherein the components include sensors or actuators.

Claim 23 (Currently Amended): A design tool, comprising:

a computer configured for design and verification of a system architecture, the system architecture including a plurality of electrical components connected to each other, the components including electronic control units, sensors, and actuators, the design tool configured to:

a) identify a set of undesirable events and ascribe to each of the undesirable events an indicator of ~~their~~ severity;

b) associate ~~where possible~~ each of the undesirable events with ~~one or more~~ any involved actuator of the actuators of the system architecture;

c) develop a functional specification of an initial architecture proposed for implementation of the system architecture, the functional specification of the initial architecture including dataflow for and between components thereof;

d) refine on the functional specification fault tolerance requirements associated with the severity of each of the undesirable events and issue refined fault tolerance requirements of the functional specification;

e) produce replicates in the functional specification together with attached indicators of ~~independence~~ freeness of the replicates from other of the replicates, the indicators reflecting the refined fault tolerance requirements;

f) define a hardware structure for the system architecture;

g) map the functional specification onto the hardware structure; and

h) verify automatically that the indicators of ~~independence~~ freeness are preserved during the mapping.

Claim 24 (Currently Amended): A ~~method~~ design tool according to claim 23, wherein the system includes a fault tolerant system.

Claim 25 (Currently Amended): A ~~computer program product~~ design tool according to claim 23, wherein the components include sensors or actuators.

Claim 26 (Currently Amended): A ~~method~~ design tool according to claim 23, wherein the hardware structure is in a form of a series of electronic control units connected to each other by networks.

Claim 27 (New): A method according to claim 10, wherein the verifying automatically that the indicators of freeness are preserved during the mapping includes verifying that a first wire which carries a first data flow and a second wire which carries a second data flow, the first data flow and the second data flow being free, are not connected to a same connector.

Claim 28 (New): A computer readable storage medium according to claim 20, wherein the verifying automatically that the indicators of freeness are preserved during the mapping includes verifying that a first wire which carries a first data flow and a second wire which carries a second data flow, the first data flow and the second data flow being free, are not connected to a same connector.

Claim 29 (New): A design tool according to claim 23, wherein, upon verifying automatically that the indicators of freeness are preserved during the mapping, the design tool verifies that a first wire which carries a first data flow and a second wire which carries a second data flow, the first data flow and the second data flow being free, are not connected to a same connector.